Adaptive Evolutionary Neural Networks for Forecasting and Trading without a Data Snooping Bias

Georgios Sermpinis, University of Glasgow, Georgios.sermpinis@glasgow.ac.uk

Thanos Verousis, University of Bath

Konstantinos Theofilatos, University of Patras

Abstract

In this paper, we present two Neural Network based techniques, an adaptive evolutionary Multilayer Perceptron (aDEMLP) and an adaptive evolutionary Wavelet Neural Network (aDEWNN). The two models are applied to the task of forecasting and trading the SPDR Dow Jones Industrial Average (DIA), the iShares NYSE Composite Index Fund (NYC) and the SPDR S&P 500 (SPY) exchange traded funds (ETFs). We benchmark their performance against two traditional MLP and WNN architectures, a Smooth Transition Autoregressive model (STAR), a moving average convergence/divergence model (MACD) and a random walk model. We show that the proposed architectures present superior forecasting and trading performance compared to the benchmarks and are free from the limitations of the traditional Neural Networks such as the data snooping bias and the time-consuming and biased processes involved in optimizing their parameters.

Keywords: Artificial Intelligence, Financial Forecasting, Data Snooping, Trading

1. INTRODUCTION

The need for forecasting techniques that are able to capture stock market patterns and produce profitable trading signals are the driving force behind numerous research papers and monographs in financial forecasting. Since the editorials of Charles Dow, several researchers and practitioners have developed forecasting methods that appear able to produce profitable trading signals. However, critics argue that these results are not genuine and they attribute the good performance of forecasting models to data snooping and to luck (see for exampleYang *et. al.* (2008) and Yang *et. al.* (2010)).

In this respect, Artificial Neural Networks (NNs) have become a popular tool in most fields of finance with a wide range of applications ranging from stock market forecasting to bankruptcy and sales prediction (see Andrawis *et al.* (2011)). Their ability to capture the discontinuities, the nonlinearities and the high complexity found in datasets such as financial time series is well documented in the literature (De Gooijer and Hyndman (2006) and Zhang *et. al.* (1998)). However, their practical limitations and the contradictory empirical evidence around their forecasting powers are creating scepticism among practitioners. The selection of their parameters, their architecture and their inputs are based to some extent on the practitioner's knowledge or on statistical algorithms (see, amongst others, White (1989), Zapranis and Refenes (1999) and Terasvirta *et. al.* (2005)) which are either only limited to the specific architecture of each network or they only search the potential space via a linear approach, which limits their effectiveness. In financial applications, these shortcomings have led practitioners to alienate NNs, as in one respect their optimization procedures require expertise and, to an extent, time-consuming experiments, whilst their forecasts usually suffer from data snooping biases.

The aim of this paper is threefold: First, we introduce two NN hybrid techniques; an adaptive evolutionary Multilayer Perceptron (aDEMLP) and an adaptive evolutionary Wavelet Neural Network (aDEWNN). Second, we demonstrate the forecasting and trading superiority of the aforementioned NNs over a set of benchmarks that dominate the relevant literature. Third, we show that the proposed models are free from any data snooping bias (an effect that might explain the contradictory empirical evidence on previous NNs applications in forecasting). Overall, the proposed models are able to exploit the superior forecasting power associated with the non-linear nature of NNs, while avoiding a complicated and objective NN training procedure.

The combination of ADE with NNs is not new. A handful of applications exist in other aspects of science (see for example, Ilonen *et.al*. (2003) and Chauhan *et.al.* (2009)). This study is the first application of hybrid NN and ADE algorithms in financial forecasting and trading. Additionally, compared to previous studies the proposed models of this study are fully adaptive. The ADE algorithm is applied in all steps of the NNs optimization and is not limited in the training procedure.

All models are applied to the task of forecasting and trading on the DIA, NYC and SPY ETFs from 2 January 2004 to 31 December 2012. Their performance is benchmarked against traditional MLP and WNN architectures, autoregressive moving average models (ARMA), a Smooth Transition Autoregressive Model (STAR) and a random walk model. In the aDEMLP and aDEWNN models, the selection of inputs, parameter optimization and architecture are generated through adaptive Differential Evolution (aDE) algorithms which require no practitioner Along similar lines, data snooping occurs when a dataset is used more than once for purposes of inference or model selection (White (2000)). In financial forecasting applications, data snooping may show superior trading performance which is only due to chance. The data snooping bias unfortunately dominates NN financial applications,

which is caused by the complexity and the large number of parameters that need to be optimized in NNs (Zhang and Hu (1998)). In the aDEMLP and aDEWNN architecture, the process of the simultaneous optimization of the NN's structure, parameters and inputs in a single optimization procedure averts the data snooping effect and excludes any bias from the estimation procedure.

Several studies have considered the data snooping effect in trading applications. Brock *et. al.* (1992) study two trading rules (a moving average and a trading range break) on the Dow Jones Index from 1897 to 1986. The authors, in order to mitigate the data snooping, utilize a very long data series and attach importance to the robustness of results across various non-overlapping sub-periods for statistical inference. Sullivan *et. al.* (1999) study 7,846 simple trading rules (filters, moving averages, trading range breaks, and channel breakouts), and argue that is possible to find profitable trading strategies that are free from data snooping on the DJIA and the S&P 500 indices. The same technical trading rules were studied by Qi and Wu (2006) for the FX market. The authors apply White's (2000) Reality Check (RC) and their results indicate that the profitability of their strategies is stronger over the period 1974 to 1985 but is relatively weak from 1986 to1998. Allen and Karjalainen (1999) use a Genetic Algorithm (GA) for the application of technical trading rules on the S&P 500 index and thus avoid the data snooping effect. Nevertheless, after considering transaction costs, their rules do not earn consistent excess returns over a simple buy-and-hold strategy. Marshall *et. al.* (2008) studied 15 major commodity futures series with over 7,000 trading rules. They found no evidence of profitability from their trading strategies after considering the data snooping bias. Finally, Harris and Yilmaz (2009) used kernel regression and a high pass filter to identify and trade the non-linear trend of several monthly exchange rates, arguing that the small set of moving average rules that they use as benchmarks, protect their conclusions from the data snooping bias.

In general, the aforementioned papers consider linear, naive technical models as trading rules. However, in more complex, non linear algorithms, the estimation procedures increase the chance of producing results that suffer from estimation biases. More specifically, in the field of Artificial Intelligence and financial forecasting, only a handful of applications consider the data snooping bias. Fernandez-Rodriguez *et. al.* (2000) forecast and trade the general index of the Madrid Stock Market with NNs. They argue that the long out-of-sample period protects their forecasts from the data snooping effect although they do not apply any formal statistical test to verify this argument. Jasic and Wood (2004) trade with NNs the S&P 500, DAX, TOPIX and FTSE stock market indices, using a large testing set and scrutinizing the results with multiple performance measures, thus reducing the exposure of the models to selection bias. However, no formal statistical test is applied to verify the validity of their results. Yang *et. al.* (2008) employ a NN and other regression techniques to examine the potential martingale behaviour of Euro exchange rates in the context of out-of-sample forecasts. Their results indicate that, while a martingale behaviour cannot be rejected for Euro exchange rates with major currencies such as the Japanese yen, British pound, and US dollar, there is nonlinear predictability in terms of economic criteria with respect to several smaller currencies (such as the Australian dollar, the Canadian dollar and the Swiss franc). In their application, the data snooping bias was considered with the RC test. Huck (2010) applies NNs and a multi-criteria decision making method in a S&P 100 stock pair trading application with positive results, Bekiros (2010) developed a hybrid neurofuzzy system which accurately forecasts the direction of the market for 10 of the most prominent stock indices of the US, Europe and Southeast Asia, while Wang *et. al.* (2012) use GA-NN hybrid models to successfully forecast the SZII index. However, all the previous three applications do not provide any formal statistical test on possible biases on their results and rely on the subjectivity of their optimization procedure. Sermpinis *et. al.* (2012) avoid the data snooping

approach with a semi adaptive NN algorithm and Yang *et. al.* (2010) study the predictability of eighteen stock indices with NNs and linear models. Their models demonstrate low predictability when the data snooping bias was considered. Gradojevic and Gencay (2013) apply a fuzzy logic algorithm to reduce the uncertainty embedded in trading strategies. The validity of their trading application is verified through a robustness check applied to their fuzzy logic model.

It is worth noting that the previous NN trading applications either discovered the data snooping bias in their results (Yang *et. al.* (2008) and Yang *et. al.* (2010)) or conducted a simple robustness check on their model, but failed to directly test for the data snooping bias (Gradojevic and Gencay (2013)). The semi-adaptive NN procedure of Sermpinis *et. al.* (2012) still requires some experimentation from the practitioner while Fernandez-Rodriguez *et. al.* (2000), Jasic and Wood (2004), Bekiros (2010), Huck (2010) and Wang *et. al.* (2012) fail to provide convincing evidence through a formal statistical test. Sullivan *et. al.* (1999) and Marshall *et. al.* (2008) explore a large universe of trading strategies which might prove a useful academic application on market efficiency, but lacks realism. This is because investors are likely to indentify a handful of profitable trading strategies in-sample and apply them out-of-sample rather than explore the utility of trading strategies that perform badly in the in-sample period. Lastly, it is worth noting the drawback of the RC test which dominates the aforementioned literature. The RC test loses power dramatically when many poor models are included in the same test (Hsu *et. al.* 2010). This is problematic in applications such as in Marshall *et. al.* (2008) when a large universe of trading strategies is explored, or when some strategies greatly underperform on their forecasting task.

The remainder of the paper is organised as follows: Section 2 describes the dataset used for this research and Section 3 provides a brief overview of NNs and explains the aDE algorithms, the aDEMLP and aDEWNN models and the benchmark models. The statistical

and trading evaluation of the proposed models against their benchmarks comprises Section 4 while Section 5 provides some concluding remarks.

2. Dataset

In this study, we apply the two NN models to the task of forecasting and trading the logarithmic returns of the DIA, NYC and the SPY ETFs, which are designed to replicate the Dow Jones Industrial Average, the NYSE Composite and S&P 500 stock market indices respectively. In general, ETFs offer investors the opportunity to trade stock market indices with very low transaction costs.[1] The summary statistics of the three return series are presented in Table 1 below.

Insert Table 1 here

As anticipated, the three returns series exhibit skewness and high kurtosis. The Jarque-Bera statistic confirms that the three return series are non-normal at the 99% confidence level.

The proposed models will be evaluated through two forecasting exercises. The periods from 2 January 2004 to 31 December 2008 and 2 January 2006 to 31 December 2010 will act as initial in-samples and the periods from 2 January 2009 to 31 December 2010 and 2 January 2011 to 31December 2012 will act as out-of-sample periods respectively. That is, the first forecasting exercise will include the effects of the recent financial crisis. The parameters of the forecasting models will be optimized during the in-sample periods and their performance will be validated in the unknown, out-of-sample datasets. This estimation will be rolled forward every three months. For example, initially the models will be trained from

[1] The transaction costs for the three ETFs tracking their respective benchmarks do not exceed 0.5% per annum for medium size investors (see, for instance, www.interactive-brokers.com).

02/01/2004 to 31/12/2008 and validated from 02/01/2009 to 31/03/2009. Then, the in-sample period will be rolled forward over three months (01/04/2004 to 31/03/2009), the parameters of the forecasting models will be re-estimated and their performance will be validated from 01/04/2009 to 31/06/2009. These rolling forward estimations are conducted eight times for each forecasting exercise. This is done because, in real world trading environments, quantitative traders update the parameters of their trading strategies frequently in order to capture any structural breaks in their estimations.

This training approach is expected to limit the effect of the current economic turbulence on our estimations and will therefore increase the performance of our models. Also, the relationship between the dependent and the independent variables in our models is allowed to time-vary and the rolling forward estimation limits the data snooping bias of our statistical benchmarks and adds further validity to the forecasting and trading exercise.

3. Forecasting Models

3.1 Neural Networks Framework

Neural networks exist in several forms in the literature. This study focuses on the Multi-Layer Perceptron (MLP), which is the most popular NN architecture, and the Wavelet Neural Network (WNN), which is a NN architecture designed to overcome convergence limitations of the MLP Neural Networks.

3.1.1 MLP

A standard MLP neural network has three layers and this setting is adapted in the present paper as well. The first layer is referred to as the input layer (the number of its nodes

corresponds to the number of explanatory variables) and the last layer is referred to as the output layer (the number of its nodes corresponds to the number of response variables). An intermediary layer of nodes which is also referred to as the hidden layer, separates the input from the output layer and its number of nodes defines the amount of complexity the model is capable of fitting. In addition, the input and hidden layer contain an extra node, called the bias node. The latter has a fixed value of one and serves the same function as the intercept in traditional regression models. Normally, each node in one layer has connections to all the other nodes in the next layer. The network processes information as follows: the input nodes contain the values of the explanatory variables. Since each node connection represents a weight factor, the information reaches a single hidden layer node as the weighted sum of its inputs. Each node of the hidden layer passes the information through a sigmoid function and on to the output layer if the estimated value is above a threshold. The training of the network (which is the adjustment of its weights in the way that the network maps the input value of the training data to the corresponding output value) starts with randomly initialized weights and proceeds by applying a learning algorithm referred to as the backpropagation of errors (Shapiro (2000)). The learning algorithm simply tries to find those weights which minimize error function. Since networks with sufficient hidden nodes are able to learn the training data (as well as their outliers and their noise) by rote it is crucial to stop the training procedure at the right time to prevent overfitting (this is called ‘early stopping’). This is usually achieved by dividing the in-sample dataset into two subsets respectively called the training and test sets which are used for simulating the available data to fit and tune the model. The network parameters are then estimated by fitting the training data using the abovementioned iterative procedure (backpropagation of errors). The iteration length is optimised by maximising a fitness function in the test dataset. Finally, the predictive value of the model is evaluated by applying it to the validation dataset (i.e. the out-of-sample dataset).

3.1.2 WNN

WNNs are a generalized form of radial basis function feed forward neural networks. Similar to simple MLPs they represent a three-layered architecture with only one hidden layer. In contrast to simple MLPs, WNNs use radial wavelets as activation functions to the hidden layer whilst using the linear activation function in the output layer. The information processing of a WNN is performed as follows. Suppose $x=[x_1,...x_d]$ to be the input signal, where d is the inputs dimensionality, then the output of its j-th hidden neuron is estimated using Equation (1).

$$\psi_j(x) = \prod_{i=1}^{d} \phi_{d_{ij},t_{i,j}}(x_i) \tag{1}$$

Where $\phi_{d_{i,j},t_{i,j}}$ is the wavelet activation function (one of Mexican Hat, Morlet and Gaussian wavelet) of the j-th hidden node and $d_{i,j}$ and $t_{i,j}$ are the scaling and translational vectors respectively. The output of the WNN is computed by estimating the weighted sum of the outputs of each hidden neuron using the weights that connect them with the output layer. The learning process involves the approximation of the scaling and translational vectors which should be used for the hidden layer and of the weights that connect the hidden layer with the output. For the approximation of the scaling vector, we used the methodology proposed by Zhang and Benveniste (1982) and for the approximation of the translational vectors, we used the k-means algorithm (Zainuddin and Pauline (2010)) with k being the number of hidden nodes. The weight vector W can easily be computed analytically by computing the following equation:

$$W=(\text{¨}\ \text{¤}\text{¨}\ )^{-1}\text{¨}\ \text{¤}Y \quad (2)$$

Where

$$\Psi = \begin{pmatrix} \psi(x_1,d_1,t_1) & \psi(x_1,d_2,t_2) & \dots & \psi(x_1,d_m,t_m) \\ \psi(x_2,d_1,t_1) & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \psi(x_d,d_1,t_1) & \psi(x_d,d_2,t_2) & \dots & \psi(x_d,d_m,t_m) \end{pmatrix} \quad (3)$$

The matrix ¨ is estimated using Equation (1) and the elements of the input, scaling and translational vectors. This deterministic method for the weights of a WNN is its stronger advantage against MLP NNs which use the backpropagation algorithm for this task and thus face the problems of low speed and becoming trapped in local optimal solutions.

3.2 Neural Network Parameterization

In order to apply a NN to a specific problem, the practitioner needs to parameterize the NN architecture. First, the numbers and the nature of the inputs need to be selected and the number of hidden nodes in the hidden layer along with the learning rate and the momentum (how fast the NN "learns") need to be chosen. Moreover, the nature of the activation functions which are to be used for the hidden and output layers should be selected. In case of sigmoid functions, the parameter which determines the gradient of the sigmoid function should also be optimized. These characteristics are crucial for the performance of a NN and its forecasting capability.

In the literature, the selection of these parameters comes is estimated using trial and error, linear statistical procedures or semi-adaptive non-linear algorithms in subsets of the in-sample period (see, amongst others, White (1989), Zapranis and Refenes (1999) and Terasvirta *et. al.* (2005)). These procedures require some knowledge of NN modelling from the practitioner (as experimentation can be unlimited) and are not widely studied. Overfitting

of the algorithms, data snooping effects and the overestimation of the results are common issues in over- and under-trained NNs. In this paper instead, we apply evolutionary optimization algorithms to optimize the parameters of MLP and WNN models. These algorithms were developed for this purpose and are designed to be fully adaptive. In order to achieve better convergence behavior, the parameter values are adapted during an evolutionary process. The adaptive nature of the proposed NN optimization procedures aims to eradicate the dangerous and time-consuming trial and error parameterization approach that leads to overfitting and data snooping problems and to introduce algorithms that require no knowledge of NN modeling from the practitioner.

3.3 Adaptive Differential Evolution

The differential evolution (DE) algorithm is currently one of the most powerful and promising stochastic real parameter optimization algorithms (Das and Suganthan (2011)). As all evolutionary algorithms, it starts with a set of randomly generated solutions to the examined optimization problem and then it iteratively applies selection, mutation and crossover operators until termination criteria are reached. In contrast to genetic algorithms, DE is mainly based on a specific mutation operator which perturbs population individuals with the scaled differences of randomly selected and distinct population members. This mutation operator provides DE with very strong exploitation properties regarding its optimization capabilities. Moreover, DE is able to handle continuous gene representation. Thus, candidate solutions are represented as strings of continuous variables comprising feature and parameter variables. In order to compute the discrete values to optimize (for example, the size of the hidden layer) the continuous values were rounded. This property of DE enables the silent evolution of its genomes as it differentiates genotype from phenotype. Thus, the values of its genes may be altered slightly without affecting the discrete values

extracted from them. An explanation for this property is that even if the continuous values which are stored in the genes of the DE algorithm are slightly altered, this does not necessarily mean that this variation with be projected in the discrete values that they encode.

Another difference between the DE and the classical GA algorithm is that the DE operators are applied in a different order. In particular, the mutation operator is initially applied to all the chromosomes of the population, then the crossover operator is implemented and finally the produced solution is subjected to a selection mechanism. This selection mechanism is not a probabilistic one as in the Roulette Wheel Selection mechanism of the GAs, but allows the new solution to become a member of the population only if it surpasses the initial solution when comparing the fitness values.

The basic flowchart of the proposed DE algorithm when applied to the optimization of the structure and the parameters of the artificial neural networks is presented in Figure 1.

*** insert Figure 1 here***

The parameters which were optimized with DE were: the size of the hidden layer, the activation functions of the hidden and output layers and the momentum and learning rates parameters. In case of optimizing the WNNs, the parameters that were optimized were the number of hidden nodes and the activation functions in this layer (as the activation function for the output layer in WNNs is always the linear one). In both cases, DE also searches for the optimal feature subset that should be used as inputs in the NNs. This pool of potential predictors (a number of variables that might be related with the forecasting problem) is consisted by the first fifteen autoregressive lags of the variables presented in Table 2.

*** insert Table 2 here***

The DE algorithm will select the optimal set of inputs for the NNs from the 180 potential predictors. The parameterization procedure is repeated each time we roll forward our estimation.

As indicated in Figure 1, after the initialization of a population of random solutions (population size equal to 30 was used), we estimated their fitness values. The fitness function (see Equation (4) below) which would be deployed for both hybrid approaches is specifically designed for financial forecasting purposes[2]. The higher the fitness (estimated by using the proposed fitness function) of a specific model, the better it is considered to be.

$$\text{Fitness} = \text{annualized return} - 10 * \text{RMSE} - 0.001 * \#\text{selected\_inputs} \quad (4)$$

The purpose of Equation (4) is twofold. First, it aims to bring a balance between financial profitability and statistical accuracy. The root mean squared error factor is multiplied by 10 so it is on the same level with the annualised return[3]. The second goal is to reduce the complexity of the extracted prediction models (see the last part of Equation (6)). The factor is multiplied by 0.001 as in this application the small algorithm complexity is of minor importance compared to the financial profitability and the statistical accuracy goals.

The next step of the proposed hybrid method is the application of the mutation and crossover operators to produce new solutions.

---

[2] The financial performance of the proposed models is slightly worse when you consider as fitness function the RMSE. The ranking of our best models in trading terms remains the same.

[3] For example for the ADE-WNN model and the DIA series during the 01/07/2004 to 31/03/2011 estimation, the annualized return's highest value found experimentally in-sample was 0.4 while the minimum value found for the RMSE was 0.05.

The mutation operator which is used in our proposed wrapper method: for every population member Xi, initially selects three random distinct members of the population ($X_{1,i}$, $X_{2,i}$, $X_{3,i}$) and produces a donor vector using the Equation (5):

$$V_i = X_{1,i} + F *(X_{2,i} – X_{3,i}) \quad (5)$$

Where F is called the mutation scale factor.

We apply a binomial crossover operator which combines every member of the population Xi with its corresponding donor vector Vi to produce the trial vector U_i, as outlined in Equation (6).

$$U_i(j) = \begin{cases} V_i(j) \quad , \quad if\,(rand_{i,j}[0,1] \le Cr) \\ X_i(j) \quad , \quad otherwise \end{cases} \quad (6)$$

Where $rand_{i,j}[0,1]$ is a uniformly distributed random number and Cr is the crossover rate.

Next, every trial vector U_i is evaluated and if it suppresses the corresponding member of the population X_i, it subsequently takes its position in the population. It is noteworthy that in contrast to other meta-heuristic approaches, the population of our approach only improves over the course of the evolutionary process following the principles of the selection operator in the Differential Evolution method.

The most important control parameters of a DE algorithm are the mutation scale factor F and the crossover rate Cr. Parameter F controls the size of the differentiation quantity which is going to be applied to a candidate solution from the mutation operator. Parameter Cr determines the number of genes which are expected to change in a population member.

Several approaches have been developed to control these parameters during the evolutionary process of the DE algorithm (see Das and Suganthan (2011)). In our adaptive DE version, we deployed one of the most recent promising approaches (Qin *et. al.* (2009)), which for every iteration, it a random value for the F parameter selected from a uniform distribution with mean of 0.5 and standard deviation of 0.3 and a random value for the parameter Cr from a uniform distribution with mean Crm and standard deviation of 0.1. Crm is initially set to 0.5. During the evolutionary process, the Crm is replaced with values that have generated successful trial vectors. Thus, this approach replaces the sensitive user-defined parameters F and Cr with less sensitive parameters, ie their mean values and their standard deviations.

The termination criterion is a combination of the maximum number of iterations and a convergence criterion. The maximum number of iterations was set to 100 and the convergence criterion is satisfied when the mean population fitness is less than 5% away from the best population fitness for more than five consecutive iterations.

3.4 Benchmarks

The performance of the aDE-MLP and aDE-WNN models is benchmarked against traditional MLP and WNN architectures, a Smooth Transition Autoregressive model (STAR), moving average convergence/divergence models (MACD), a naïve strategy and a random walk model.

The random walk model takes the form of the sum of the in-sample mean plus an error term that follows the standard normal distribution. The MACD model combines two moving averages with different moving average lengths. Positions are taken if the moving averages intersect. If the short-term moving average intersects the long-term moving average from below, a ‘long’ position is taken. Conversely, if the long-term moving average is intersected

from above, a 'short' position is taken.[4] We select the MACD models that present the highest trading performance in the in-sample period. STARs, initially proposed by Chan and Tong (1986) are extensions of the traditional autoregressive models (ARs). The STAR combines two AR models with a function that defines the degree of non-linearity (smooth transition function). The general two-regime STAR specification is the following:

$$\hat{Y}_t = \Phi_1' \mathrm{X}_t (1 - F(z_t, \zeta, \lambda)) + \Phi_2' \mathrm{X}_t F(z_t, \zeta, \lambda) + u_t \tag{7}$$

Where:

- $\hat{Y}_t$ the forecasted value at time *t*
- $\Phi_i = (\tilde{\varphi}_{i,0}, \tilde{\varphi}_{i,1}, ...\tilde{\varphi}_{i,p}), i = 1,2$ and $\tilde{\varphi}_{i,0}, \tilde{\varphi}_{i,1}, ...\tilde{\varphi}_{i,p}$ the regression coefficients of the two AR models
- $\mathrm{X}_t = (1, \tilde{\chi}_t')'$ with $\tilde{\chi}_t' = (Y_{t-1}, ..., Y_{t-p})$
- $0 \leq F(z_t, \zeta, \lambda) \leq 1$ the smooth transition function
- $z_t = Y_{t-d}, d > 0$ the lagged endogenous transition variable
- ¶ the parameter that defines the smoothness of the transition between the two regimes
- » the threshold parameter
- u_t the error term

In this paper, we follow the steps of Lin and Terasvitra (1994) in order to determine when the series is best modeled as a Logistic STAR or an Exponential STAR process. The

[4]A 'long' position implies buying the ETF at the current price, while a 'short' position implies selling the ETF at the current price.

specification of the STAR and the MACD models are determined in the in-sample period. These specifications are updated every three months by rolling forward the estimations.

In addition to the above technical and statistical models, the aDE-MLP and aDE-WNN are benchmarked against traditional and semi adaptive MLP and WNN. The traditional MLP and WNN models are trained with the trial and error approach that is described in section 3.2 and dominates the finance literature (see amongst others Fernandez-Rodriguez *et. al.* (2000), Jasic and Wood (2004), Wang *et. al.* (2012 and Sermpinis *et. al.* (2012)). The semi adaptive MLP and WNN models apply the aDE algorithm only to select the appropriate set of the NN's inputs (hereafter s-MLP and s-WNN). The rest of the NN parameters are optimized with the usual trial and error approach (using the in-sample dataset). Such semi-adaptive NN (with less efficient optimization algorithms) approaches have recently be presented in the literature with promising results (see Mingming and Jinliang (2012) and Sermpinis *et. al.* (2012)).

4. Empirical Results

4.1 Statistical Performance

In Table 3, we present the statistical performance of all models for the out-of-sample period.[5] For the Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Theil-U statistics, the lower the output, the better the forecasting accuracy of the model concerned. The Pesaran-Timmermann (PT) test (1992) examines whether the directional movements of the real and forecast values are in step with one another. The null hypothesis is that the model under study has no power on forecasting the ETF return series. The HLN statistic suggested by Harvey *et. al.* (1997) has as null hypothesis the equivalence in forecasting accuracy between two forecasting models. The HLN test is applied in order to verify that a forecasting

[5] The statistical ranking of our models is identical in the in-sample and out-of-sample sub-periods. The in-sample statistical performance of our models is not presented for the sake of space and is available upon request.

model is statistically different to the RW. The PT test follows the standard normal distribution, while the HLN test follows the Student's t -distribution with n-1 degrees of freedom (where n is the number of forecasts).

*** insert Table 3 here***

From Table 3, we note that the NNs architectures clearly outperform their benchmarks for all the statistical measures retained. The MLP architectures seem slightly more accurate for the DIA returns, while the WNN models for the NYC and SPY series. The PT statistics demonstrate the inability of the RW, MACD and STAR models to accurately forecast the directional change of the three series under study. The HLN statistics indicate that our NN forecasts are statistically different to an RW model at the 1% confidence level. In contrast, the MACD and the ARMA models present a disappointing performance as they seem unable to outperform the RW benchmark. In general, the statistical accuracy of our models is worse in the first forecasting exercise.

4.2 Trading Performance

In Table 4, we present the trading performance of our models[6] after transaction costs while their in-sample performance is presented in Appendix A.1.

Insert Table 4 here

We note that after transaction costs, the aDEWNN model outperforms its benchmark in terms of annualized return and information ratio. aDEWNN is closely followed by the WNN and

[6] The computational of cost of our proposed models (ADEMLP and ADEWNN) is approximately ten minutes with a modern PC for a single simulation. In our simulations, we used Matlab R2010a and a PC with Intel Processor I7 and 4GB RAM.

the s-WNN algorithms with drop of 1.5% of the average annualized return. The MLP models seem unable to produce profitable signals with the same magnitude as the WNNs and present smaller trading performance. The profitability of the RW, STAR and MACD models is disappointing with information ratios that do not exceed 0.38 after transaction costs. The maximum drawdown (the essence of risk for an investor in financial markets) of the NNs models is considerably lower than their statistical/technical benchmarks. In line with the statistical performance of our models, we note that the trading results of the new models have substantially improved in the second forecasting exercise. This can be explained by the uncertainty that dominated the financial markets during 2009 and 2010. It was therefore anticipated that this uncertainty would affect our results and worsen the forecasting performance of our models.

4.3 Data Snooping

In trading applications, it is important to be able to identify genuine profitable trading strategies as model profitability may be attributed to luck and its performance might not be generalized. This bias is difficult to avoid in complicated non-linear models such as in NNs where their parameterization requires extensive experimentation. In order to test if our proposed models forecasts suffer from data snooping, we created four groups of forecasting models. The first group includes all nine models under study. The second group includes the simple MLP and WNN architectures and the RW, MACD and STAR models. The third group consists of the s-MLP, s-WNN, RW, MACD and STAR models while the fourth group is made up of the aDEMLP and the aDEWNN models and the three technical/statistical benchmarks. In order to test for the data snooping bias we apply the Hansen (2005) test (SPA) to the four groups. The SPA test is built upon the RC test and the implementation of the two tests is similar. The SPA test is less conservative as it avoids the least favourable

configuration, i.e., the configuration that is least favourable to the alternative (Hsu *et. al.* (2010)). In other words, the SPA test is more powerful and less sensitive to poor and irrelevant alternatives. The null hypothesis is that the benchmark is not inferior to any alternative forecast. As a benchmark we use a buy-and-hold strategy. For more details on the SPA test see Hansen (2005). As a performance measurement to the SPA test we apply the Sharpe ratio while the characteristics of our SPA test are the same as in Neuhierl and Schlusche (2011).[7] The SPA p-values for the out-of-sample period are presented in Table 5 below.

Insert Table 5 here

From Table 5, we note that nominal p-values are significant for all groups suggesting that the best-performing rules significantly outperform the buy-and-hold strategy. However, after correcting for data snooping, the results are insignificant for the groups that contain the traditional and the semi-adaptive NNs. For the fourth group, that contains the two fully adaptive NNs and the three statistical/technical benchmarks, the results remain significant at the 5% confidence level for all three return series.

5. Conclusions

In this paper, a data snooping-free computational framework which deploys an adaptive Differential Evolution approach to optimize the MLP and WNN parameters and their inputs was introduced. Compared to the limited literature of hybrid NN and ADE models, this is their first application in financial forecasting and trading while the proposed models are fully adaptive.

[7] The smoothing parameter is set to 0.5 and the number of bootstraps to 1000.

The experimental results proved that the proposed models provided profitable strategies which outperformed, in terms of sharp ratio, a buy-and-hold strategy even when the data snooping effect was corrected. In contrast, classical NN and semi adaptive approaches failed to significantly outperform the benchmark models when the data snooping effect was corrected even if they outperformed it initially. Comparing the WNN based approaches with the MLP, the former outperformed the latter. This finding may be attributed to their simpler training algorithm which is not based to the backpropagation procedure that suffers from overfitting and probabilities of becoming trapped in local optimal solutions.

These results should contribute towards convincing academics and practitioners of the utility of complex non linear regression models, such as hybrid NNs, in finance, forecasting and trading. The introduced algorithm transforms the NN optimization into a fully adaptive procedure capable of providing accurate forecasts free of the biases that dominate the related literature.

References

Allen, F. and Karjalainen, R. (1999) 'Using genetic algorithms to find technical trading rules', *Journal of Financial Economics*, 51, 2, 245-271

Andrawis, R.R., Atiya, A.F. and El-Shishiny, H. (2011) 'Forecast combinations of computational intelligence and linear models for the NN5 time series forecasting competition', *International Journal of Forecasting*, 27, 3, 672–688

Bekiros, S. D. (2010), 'Fuzzy adaptive decision-making for boundedly rational traders in speculative stock markets', *European Journal of Operational Research*, 202, 1, 285-293

Brock, W., Lakonishok, J. and LeBaron, B. (1992), 'Simple technical trading rules and the stochastic properties of stock returns', *The Journal of Finance*, 47, 1731–1764

Chan, K.S. and Tong, H. (1986), 'On estimating thresholds in autoregressive models', *Journal of Time Series Analysis*, 7, 178-190

Chauhan, N., Ravi, V. and Karthik Chandra, D. (2009), 'Differential evolution trained wavelet neural networks: Application to bankruptcy prediction in banks', *Expert Systems with Applications*, 36, 4, 7659-7665.

Das, S. and Suganthan, P. (2011), 'Differential Evolution: A survey of the State-of-the-Art', *IEEE* Transactions on evolutionary computation. 15, 1, 4-30

De Gooijer, J.G. and Hyndman, R.J. (2006), '25 years of time series forecasting', *International Journal of Forecasting*, 22, 3, 443-473.

I huqdqgh}-

Gradojevic, N. and Gençay, R. (2013), 'Fuzzy logic, trading uncertainty and technical trading', *Journal of Banking and Finance*, 37, 2, 578-586

Ilonen, J., Kamarainen, J. K., & Lampinen, J. (2003), 'Differential evolution training algorithm for feed-forward neural networks', *Neural Processing Letters*, 17, 1, 93-105.

Hansen, P. R. (2005), 'A test for superior predictive ability', *Journal of Business and Economic Statistics*, 23, 365–380.

Harris, R. and Yilmaz, F. (2009), 'A momentum trading strategy based on the low frequency component of the exchange rate', *Journal of Banking and Finance*, 33, 9, 1575-1585

Harvey, D., Leybourne, S. and Newbold, P. (1997), 'Testing the equality of prediction mean squared errors', *International Journal of Forecasting*, 13, 2,281–291.

Hsu, P., Hsu, Y. and Kuan, C. (2010), 'Testing the predictive ability of technical analysis using a new stepwise test without data snooping bias', *Journal Empirical Finance*, 17,3, 471-484

Huck, N. (2010). 'Pairs Ttrading and Outranking: The Multi-Step-Ahead Forecasting Case', *European Journal of Operational Research*, 207, 3, 1702-1716

Lin, C.J. and Terasvirta T. (1994) 'Testing the Constancy of Regression Parameters against Continuous Structural Changes', *Journal of Econometrics*, 62, 211-228

Lisboa, P. and Vellido, A. (2000), 'Business Applications of Neural Networks', vii-xxii, in P. Lisboa, B. Edisbury and A. Vellido [eds.] *Business Applications of Neural Networks: The State-of-the-Art of Real-World Applications*, World Scientific, Singapore

Marshall, B., Cahan, R. and Cahan, J. (2008), 'Can commodity futures be profitably traded with quantitative market timing strategies?', *Journal of Banking and Finance*, 32, 9, 1810-1819

Mingming, T. and Jinliang, Z. (2012), 'A multiple adaptive wavelet recurrent neural network model to analyze crude oil prices', *Journal of Economics and Business*, 64, 4, 256-286

Neuhierl, A. and Schlusche, B. (2011), 'Data Snooping and market-timing rule performance', *Journal of Financial Econometrics*, 9, 3, 550-58

Jasic, T. and Wood, D. (2004), 'The profitability of daily stock market indices trades based on neural network predictions: case study for the S&P 500, the DAX, the TOPIX and the FTSE in the period 1965–1999', *Applied Financial Economics*, 14, 4, 285-297

Qi, M. and Wu, Y. (2006), 'Technical trading-rule profitability, data snooping, and reality check: Evidence from the foreign exchange market', *Journal of Money, Credit and Banking*, 38, 2135–2158

Qin, A., Huang, V. and Suganthan, P. (2009), 'Differential evolution algorithm with strategy adaptation for global numerical optimization', *IEEE Transactions of Evolutionary Computation.*, 13, 2, 398-417

Sermpinis, G., Theofilatos, K., Karathanasopoulos, A., Georgopoulos, E. and Dunis, C. (2012), 'Forecasting foreign exchange rates with adaptive neural networks using radial-basis functions and Particle Swarm Optimization', *European Journal of Operational Research*, 225, 3, 528–540

Sullivan, R., Timmermann, A. and White, H. (1999) 'Data-Snooping, Technical Trading Rule Performance, and the Bootstrap', *The Journal of Finance*, 54, 5, 1647-1691

Terasvirta, T., Van Dijk, D., and Medeiros, M. (2005) 'Linear Models, Smooth Transition Autoregressions, and Neural Networks for Forecasting Macroeconomic Time Series: A Re-examination,' *International Journal of Forecasting*, 21, 755–774

Yang, J., Su, X. and Kolari, J. W. (2008), 'Do Euro Exchange Rates Follow a Martingale? Some Out-of-Sample Evidence', *Journal of Banking & Finance*, 32, 729 - 740

Yang, J., Cabrera, J. and Wang, T. (2010), 'Nonlinearity, Data-Snooping, and Stock Index ETF Return Predictability', *European Journal of Operational Research*, 200, 2, 498-507

Zapranis, A. and Refenes, P. (1999), '*Principles of Neural Model Identification, Selection and Adequacy: With Applications to Financial Econometrics*', Springer-Verlag, Berlin

Zhang, Q. and Benveniste, A. (1982), 'Wavelet networks', *IEEE Transactions on Neural Network*. 3, 889--898

Zhang, G., Patuwo, B. and Hu, M. (1998), 'Forecasting with artificial neural networks: The state of the art', *International Journal of Forecasting*, 14, 1, 35-62

Zhang, G. and Hu, M. (1998), 'Neural network forecasting of the British Pound/US Dollar exchange rate', *Omega*, 26, 4, 495-506

Zainuddin, Z. and Pauline, O. (2010), 'Improved Wavelet Neural Networks for Early Cancer Diagnosis Using Clustering Algorithms', *International Journal of Information and Mathematics Sciences*, 6, 1, 30--36

Wang, J-J., Wang, J-Z., Zhang, Z-G. and Guo, S-P. (2012), 'Stock index forecasting based on a hybrid model', *Omega*, 40, 6, 758-766

White, H. (1989), 'Learning in Neural Networks: A Statistical Perspective', *Neural Computation*, 1, 425–464

White, H. (2000), 'A reality check for data snooping', *Econometrica*, 68, 1097–1126